

WEST[Help](#)[Logout](#)[Interrupt](#)
[Main Menu](#) | [Search Form](#) | [Posting Counts](#) | [Show S Numbers](#) | [Edit S Numbers](#) | [Preferences](#) | [Cases](#)

Search Results -

Terms	Documents
L2 and Runtime	14

Database:

- US Patents Full-Text Database
- US Pre-Grant Publication Full-Text Database
- JPO Abstracts Database
- EPO Abstracts Database
- Derwent World Patents Index
- IBM Technical Disclosure Bulletins

Search:

[Refine Search](#)

Search History

DATE: Wednesday, March 19, 2003 [Printable Copy](#) [Create Case](#)
Set Name Query

side by side

DB=USPT; PLUR=NO; OP=OR

Set Name	Count	Query
<u>L3</u>	14	L2 and Runtime
<u>L2</u>	39	L1 AND instrumentation
		(6158049 6499137 6430741 6106571 5847953 5920479 5987250 6126330 5909578 5937191 5903758 6327700 6149318 6219825 6219825 6311327 5528753 6026236 5606657 5812828 5832271 6460178 6102965 6192511 5758183 5761729 5787480 5802585 5933640 5630164 5784275 5905649 5949972 6035426 5551040 5822787 4782463 5797014 5835743 5991871 6003095 6026235 6047362 6483911 5546577 5680615 5758074 5778377 5768592 4819233).pn.
<u>L1</u>	49	

END OF SEARCH HISTORY

PLVS Review

WEST Search History

DATE: Wednesday, March 19, 2003

Set Name Query

side by side

DB=USPT; PLUR=NO; OP=OR

L3 L2 and Runtime

L2 L1 AND instrumentation

(6158049 6499137 6430741 6106571 5847953 5920479 5987250
6126330 5909578 5937191 5903758 6327700 6149318 6219825
6219825 6311327 5528753 6026236 5606657 5812828 5832271
6460178 6102965 6192511 5758183 5761729 5787480 5802585
5933640 5630164 5784275 5905649 5949972 6035426 5551040
5822787 4782463 5797014 5835743 5991871 6003095 6026235
6047362 6483911 5546577 5680615 5758074 5778377 5768592
4819233).pn.

Hit Count Set Name result set

14 L3

39 L2

49 L1

END OF SEARCH HISTORY

WEST[Help](#)[Logout](#)[Interrupt](#)[Main Menu](#) [Search Form](#) [Posting Counts](#) [Show S Numbers](#) [Edit S Numbers](#) [Preferences](#) [Cases](#)**Search Results -**

Terms	Documents
L4 AND (endpoint Or (end ADJ point))	5

Database:

Search:

Search History

DATE: Wednesday, March 19, 2003 [Printable Copy](#) [Create Case](#)

[Set Name](#) [Query](#)
side by side

[Hit Count](#) [Set Name](#)
result set

DB=USPT; PLUR=NO; OP=OR

<u>L5</u>	L4 AND (endpoint Or (end ADJ point))	5	<u>L5</u>
<u>L4</u>	((717/157 717/158)!.CCLS.)	144	<u>L4</u>
<u>L3</u>	L2 and Runtime	14	<u>L3</u>
<u>L2</u>	L1 AND instrumentation (6158049 6499137 6430741 6106571 5847953 5920479 5987250 6126330 5909578 5937191 5903758 6327700 6149318 6219825 6219825 6311327 5528753 6026236 5606657 5812828 5832271 6460178 6102965 6192511 5758183 5761729 5787480 5802585 5933640 5630164 5784275 5905649 5949972 6035426 5551040 5822787 4782463 5797014 5835743 5991871 6003095 6026235 6047362 6483911 5546577 5680615 5758074 5778377 5768592 4819233).pn.	39	<u>L2</u>
<u>L1</u>		49	<u>L1</u>

END OF SEARCH HISTORY

WEST[Generate Collection](#)[Print](#)**Search Results - Record(s) 1 through 5 of 5 returned.**

1. Document ID: US 6282707 B1

L5: Entry 1 of 5

File: USPT

Aug 28, 2001

US-PAT-NO: 6282707

DOCUMENT-IDENTIFIER: US 6282707 B1

TITLE: Program transformation method and program transformation system

DATE-ISSUED: August 28, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Isozaki, Hiroko	Tokyo			JP

US-CL-CURRENT: 717/157; 717/159

ABSTRACT:

A program transformation method for transforming a source program described by a programming language into an object program described by a language executable by a data processing system, includes a process of transforming at least a part of procedure, function or sub-routine used in the source program into a form so that the object program can be stored in an arbitrary storage region of a primary storage device of the data processing system, a process of arranging procedure, function or sub-routine transformed or not transformed in the first process in the storage region corresponding to cache line of a cache memory among storage region of the primary storage device without causing cache conflict on the basis of information relating to the procedure, function or sub-routine obtained during a process of transformation of the source program into the object program, and a process of generating the object program, on the basis of the result of arrangement.

23 Claims, 19 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 16

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Sequences](#) | [Attachments](#) | [Claims](#) | [KOMC](#) | [Drawn Desc](#) | [Image](#)

2. Document ID: US 5889999 A

L5: Entry 2 of 5

File: USPT

Mar 30, 1999

US-PAT-NO: 5889999

DOCUMENT-IDENTIFIER: US 5889999 A

TITLE: Method and apparatus for sequencing computer instruction execution in a data processing system

DATE-ISSUED: March 30, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Breternitz, Jr.; Mauricio	Austin	TX		
Smith; Roger A.	Austin	TX		

US-CL-CURRENT: 717/158; 712/201, 712/214, 712/233, 712/236, 712/237, 714/35, 714/45

ABSTRACT:

A method and apparatus for sequencing computer instructions in memory (24) to provide for more instruction efficient execution by a central processing unit (CPU) (22) begins by executing the computer instructions via the CPU (22) and creating a trace file (FIG. 2) in memory (24). The trace file is then scanned using a window size greater than two (i.e., more than two instructions or basic blocks/ groups of instructions are selected as each window) and correlations are determined between several pairs of instructions in each window (FIGS. 9 and 10). The correlations obtained by the window procedure are then analyzed (FIG. 11) to determine an efficient ordering of computer instructions for subsequent execution by any target CPU.

33 Claims, 43 Drawing figures
 Exemplary Claim Number: 1
 Number of Drawing Sheets: 28

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Sequences](#) | [Attachments](#) | [Claims](#) | [KWM](#) | [Drawn Desc](#) | [Image](#)

3. Document ID: US 5802585 A

L5: Entry 3 of 5

File: USPT

Sep 1, 1998

US-PAT-NO: 5802585

DOCUMENT-IDENTIFIER: US 5802585 A

TITLE: Batched checking of shared memory accesses

DATE-ISSUED: September 1, 1998

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Scales; Daniel J.	Palo Alto	CA		
Thekkath; Chandramohan A.	Palo Alto	CA		

US-CL-CURRENT: 711/154; 711/147, 717/158

ABSTRACT:

In a distributed shared memory computer system a plurality of workstations are connected to each other by a network. Each workstation includes a processor, a memory having addresses, and an input/output interface connected to each other by a bus. The input/output interfaces connect the workstations to each other by the network. In a software implemented method for batching access checks to shared data stored in the memories, a set of the addresses of the memories are designated virtual shared addresses to store shared data. A portion of the virtual shared addresses is allocated to store a shared data structure as one or more lines accessible by instructions of the programs executing in any of the processors, the size of each line being a predetermined number of bytes. The programs are analyzed to locate a set of instructions of a particular program which access a range of target addresses storing shared data, the range of target addresses being no greater than the size of one line. In response to determining that accesses to the shared data stored at a first and last addresses of the range of target addresses are valid, the set of instructions are executed. If accesses to the shared data stored at the first or last addresses of the range of target addresses are invalid, miss handling is executed before executing the set of instructions.

26 Claims, 14 Drawing figures
Exemplary Claim Number: 17
Number of Drawing Sheets: 14

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Sequences](#) | [Attachments](#) | [Claims](#) | [KIMC](#) | [Draw Desc](#) | [Image](#)

4. Document ID: US 5530866 A

L5: Entry 4 of 5

File: USPT

Jun 25, 1996

US-PAT-NO: 5530866

DOCUMENT-IDENTIFIER: US 5530866 A

TITLE: Register allocation methods having upward pass for determining and propagating variable usage information and downward pass for binding; both passes utilizing interference graphs via coloring

DATE-ISSUED: June 25, 1996

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Koblenz; Brian D.	Seattle	WA		
Callahan, II; Charles D.	Mercer Island	WA		

US-CL-CURRENT: 717/144; 717/158, 717/159

ABSTRACT:

The present invention provides methods for allocating physical registers within a compiler phase to achieve efficient operation of a target CPU. The methods of the present invention allocate variables between physical registers and memory to accommodate local as well as global code structure. Such methods facilitate the location of variables that are heavily accessed at a portion of the code in a physical register during the execution thereof.

34 Claims, 4 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 3

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Sequences](#) | [Attachments](#)

| [KIMC](#) | [Draw Desc](#) | [Image](#)

5. Document ID: US 5142679 A

L5: Entry 5 of 5

File: USPT

Aug 25, 1992

US-PAT-NO: 5142679

DOCUMENT-IDENTIFIER: US 5142679 A

TITLE: Method and apparatus for collecting execution status data of structured program

DATE-ISSUED: August 25, 1992

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Okaki; Takashi	Hitachi			JP
Hayashi; Toshihiro	Hitachi			JP

US-CL-CURRENT: 717/151; 714/38, 717/154, 717/158

ABSTRACT:

Method and apparatus provide for collecting execution status data in the execution of an object program. In compiling a source program described in a structured format to an object program, a PROBE instruction for starting a data collecting program for collecting the execution status data, with a block identification number in an operand field thereof, is inserted at a position of an exit sentence indicating an exit of the program block. When the PROBE instruction is registered in an instruction execution register of an execution unit during a period of the execution of the object program, the data collecting program is started to collect the execution status data, which is then stored in a store table.

9 Claims, 27 Drawing figures

Exemplary Claim Number: 1

Number of Drawing Sheets: 12

[Full](#) | [Title](#) | [Citation](#) | [Front](#) | [Review](#) | [Classification](#) | [Date](#) | [Reference](#) | [Sequences](#) | [Attachments](#)

[KUMC](#) | [Draw Desc](#) | [Image](#)

[Generate Collection](#)

[Print](#)

Terms	Documents
L4 AND (endpoint Or (end ADJ point))	5

Display Format: [REV](#) [Change Format](#)

[Previous Page](#) [Next Page](#)